

Frequently Asked Questions About Callgraph on Linux*

Using the VTune™ Performance Analyzer

Rev 0.64

Topics

1. [Supported platforms and system requirements](#)
2. [Overview of remote callgraph](#)
3. [Stopping a callgraph application](#)
4. [Using user-defined signals to stop callgraph application](#)
5. [Location of callgraph temporary files](#)
6. [Duration of a callgraph session](#)
7. [Environment settings for callgraph application](#)
8. [Callgraph on multiple applications simultaneously](#)
9. [Multiple users running callgraph simultaneously](#)

Legal

[Disclaimer and Legal Information](#)

1. Supported platforms and system requirements

What platforms is callgraph supported on? Are there any restrictions?

The following platforms, operating systems, and compilers support callgraph RDC:

On IA-32 processors

- Red Hat* releases: Red Hat 7.2, Red Hat 7.3, Red Hat 9 + patches, Fedora* core 1, Advanced Server 2.1, EL 3.0
- SuSE* releases: SuSE 8.0, SuSE 8.1, SuSE 8.2, SuSE Linux Enterprise Server 8, and SuSE 9

On Itanium(R) processor family

- Red Hat* releases: Advanced Server 2.1, Enterprise Linux* 3.0
- SuSE* releases: SuSE Linux Enterprise Server 8

The above operating systems require either GLIBC version 2.2.2, 2.2.4, 2.2.5 or 2.3.2.

Applications built using the following compilers

- GCC* 2.96
- GCC 3.2
- Intel(R) compiler 7.0
- Intel(R) compiler 7.1

The callgraph run creates Unix pipes and uses file locking on the Linux filesystem. Therefore the Linux filesystem on which the remote callgraph cache directory resides must support these operations. It is recommended to use a local disk for the callgraph cache directory on the Linux system.

NFS automount utility (also known as amd daemon) may unmount directories specified in the callgraph collection but not used continuously during the run. This may cause unexpected behavior and the VTune™ Performance Analyzer may fail to run or generate results. If automounted directories are used, then they should not be unmounted while the callgraph session is running.

Callgraph results are collected and transferred to the VTune analyzer running on Microsoft* Windows*. These results are stored in memory on the Windows side. In some cases the required memory is very big and may even exceed the physical available memory. Users can limit the amount of memory used during the run in the Configure -> Options -> Call Graph settings dialog box. In the beta version the maximum callgraph results size is limited to about 800Mb. It is recommended to define virtual memory as large as possible.

There is no callgraph support for static Linux applications that have no symbols.

Note: In order to use call graph on Red Hat* 9, install the following patches:

One of the following kernels:

- kernel-2.4.20-20.9.i686.rpm
- kernel-smp-2.4.20-20.9.i686.rpm
- kernel-bigmem-2.4.20-20.9.i686.rpm

GLIBC

- glibc-2.3.2-27.9.i686.rpm
- nptl-devel-2.3.2-27.9.i686.rpm
- glibc-common
- glibc-debug
- glibc-utils

You can download these patches at <https://rhn.redhat.com/>

Please see the release notes provided with the VTune analyzer install package for further issues.

2. Overview of remote callgraph

How does remote callgraph work?

Remote callgraph works in the following way:

1. In the VTune analyzer, a user specifies the Linux application on which to perform callgraph, the desired callgraph instrumentation level, and other configuration options.
2. The VTune analyzer forwards this information to the remote callgraph collector running on the Linux system.
3. The callgraph collector instruments the closure of the application (i.e., all libraries used by the application, along with the application itself) on the Linux system. This produces a new set of binaries, representing the instrumented application, in the remote Linux callgraph cache directory (by default, /tmp/vtune_cache).
4. The instrumented application is run. During its execution, callgraph data is collected.
5. Upon successful completion, data generated during the callgraph collection is transferred to the VTune analyzer for further processing and display.

3. Stopping a callgraph application

How can I stop a callgraph application?

A callgraph application that is currently running can be stopped in several ways:

1. Normally, pressing the red stop button in the VTune analyzer should stop the callgraph application.
2. If that fails, then press the red stop button 3 times.
3. If that fails, then the process will need to be manually killed (e.g., using "`kill -9`").

Please note that callgraph applications that are abnormally stopped will *not* have their callgraph results displayed.

4. Using user-defined signals to stop callgraph application

The set of signals used by callgraph conflict with signal handling in my application. How can I resolve this?

One of the methods that stop a running application is by sending a special signal to the application's processes. The default signal is SIGUSR2 but if this signal is already used by the application, the user has to define alternative signal number by setting the new signal number in the `__Bistro_Exit_Signal__` environment variable. This variable should be set prior to running `vtserver`.

5. Location of temporary callgraph files

Where are temporary callgraph files stored on the Linux side?

There are two (potentially different) directories used on the Linux system to store temporary callgraph data. The first is the callgraph cache directory, which defaults to `/tmp/vtune_cache` (this can be changed in the VTune analyzer by going to Configure->Options->Call Graph->Collector->Cache Directories). The second is the `vtserver` data directory (see "`vtserver --help`" for details). Both directories must be writable by the user running `vtserver`.

6. Duration of a callgraph session

Can I set the duration for a callgraph session?

No. Currently, the duration setting is ignored by the callgraph collector.

7. Environment settings for callgraph application

How can I pass environment variable settings to a callgraph application?

There are two ways to pass environment variables to a callgraph application.

- Prior to starting `vtserver`, set any environment variables required by the application. Then start `vtserver`.
- Prior to launching the callgraph application, go to the Advanced Activity Configuration dialog box and under Application/Module Profiles, select the application and then press the Configure button. This will bring up a dialog box called General. Press the Advanced button and then unselect the Use default environment checkbox to enter environment variables of the form "VARIABLE=value" (e.g., "FOO=3"). Press OK to save your settings.

Note: Currently, callgraph cannot be performed on scripts. Thus, any environment variables required by the application must be set in one of the ways described above.

8. Callgraph on multiple applications simultaneously

Can I do callgraph on multiple (different) applications simultaneously?

No. Currently, only one application may have callgraph performed on it during a callgraph session. This single application must be specified in the application's module of interest field in the Application/Module Profile Configuration dialog box.

9. Multiple users running callgraph simultaneously

Can multiple users on the same Linux system run their own callgraph sessions simultaneously?

No. Currently, multiple users on same Linux system cannot perform callgraph at the same time. Users should coordinate amongst themselves to run their callgraph sessions at different (non-overlapping) times on the same Linux system.

Disclaimer and Legal Information

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT,

COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

This *FAQ document for Call graph* as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel SpeedStep, Intel Thread Checker, Celeron, Dialogic, i386, i486, iCOMP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetStructure, Intel Xeon, Intel XScale, Itanium, MMX, MMX logo, Pentium, Pentium II Xeon, Pentium III Xeon, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright© Intel Corporation 2000 - 2003.